

In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1 1. (Currently Amended) A digital system having a processor
2 comprising a processor pipeline with a plurality of pipeline
3 stages, a plurality of protected resources connected to receive
4 data from certain ones of the plurality of pipeline stages and a
5 pipeline protection mechanism, wherein the pipeline protection
6 mechanism comprises:

7 a set of shadow registers;

8 interlock circuitry for anticipating a particular access
9 conflicts for each protected resource of the plurality of protected
10 resources between the pipeline stages, an output of the interlock
11 detection circuitry being controllably connected to the set of
12 shadow registers, the interlock circuitry comprising a stall vector
13 filter, wherein the stall vector filter has a plurality of select
14 filter stages connected in a serial manner such that each of the
15 select filter stages is associated with a corresponding protected
16 resource; and

17 the set of shadow registers being interconnected with the
18 processor pipeline such that a particular data item from a first
19 pipeline stage can be redirected from a protected resource into a
20 selected shadow register only in response to an the particular
21 access conflict anticipated by the interlock circuitry so that a
22 resource access conflict is resolved without stalling the processor
23 pipeline.

1 2. (Currently Amended) The ~~processing engine~~ digital system
2 according to Claim 1, wherein the interlock circuitry comprises:

3 interlock detection circuitry operable to anticipate access
4 conflicts for all of the protected resources and the stall vector

5 filter operable to form a stall vector signal indicative of
6 anticipated access conflicts;

7 reservation and stall vector filtering circuitry connected to
8 receive the stall vector signal and operable to select an available
9 shadow register from the set of shadow registers in response to the
10 stall vector signal; and

11 shadow management circuitry connected to the reservation and
12 filtering circuitry, the shadow management circuitry having an
13 output signal controllably connected to the set of shadow
14 registers.

1 3. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 1, wherein the interlock circuitry comprises
3 arbitration circuitry for each protected resource such that each
4 arbitration circuit is definable as a specific form of a single,
5 generic arbitration function.

1 4. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 1, including pipeline control logic for
3 controlling the stages of the pipeline, the pipeline control logic
4 being connected to receive the stall control signals output from
5 the interlock circuitry based upon a result of arbitration between
6 resources.

1 5. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 1, wherein at least one resource is selected
3 from a group consisting of: a group of registers; a register; a
4 field of a register; and a sub-field of a register.

6. (Canceled)

1 7. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 1, wherein the set of shadow registers is
3 interconnected with the processor pipeline with multiplexing
4 circuitry operable to redirect a read from a protected resource to
5 a selected shadow register.

8 to 12. (Canceled)

1 13. (Previously Amended) The digital system of Claim 1 being
2 a cellular telephone, further comprising:
3 an integrated keyboard connected to the processor via a
4 keyboard adapter;
5 a display, connected to the processor via a display adapter;
6 radio frequency (RF) circuitry connected to the processor; and
7 an aerial connected to the RF circuitry.

14 to 18. (Canceled)

1 19. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 1, wherein the processor pipeline is operable to
3 update periodically, and wherein the interlock circuitry is
4 operable to anticipate access conflicts for each protected resource
5 during each pipeline period prior to each periodic update of the
6 processor pipeline.

1 20. (Currently Amended) The ~~processing-engine~~ digital system
2 according to Claim 19, wherein certain of the plurality of pipeline
3 stages are subject to access conflicts, and wherein the interlock
4 circuitry is connected to receive resource access signals provided
5 by each pipeline stage that is subject to access conflicts, whereby
6 additional control storage circuitry is not required for storing
7 conflict control information.

1 21. (New) A digital data processor including a processor
2 pipeline with a plurality of pipeline stages and operable upon
3 instructions specifying operands from a plurality of logical
4 registers comprising:

5 a plurality of physical data registers, each corresponding to
6 fixed respective logical register;

7 a set of shadow registers;

8 an interlock detection circuit for anticipating write after
9 read (WAR), write after write (WAW) and read after write (RAW)
10 register access conflicts of instructions between the pipeline
11 stages, said interlock detection circuit generating a stall vector
12 upon detection of a conflict indicating a number of pipeline stages
13 of stall for the detected conflict; and

14 a register control circuit connected to the plurality of
15 physical registers, the set of shadow register and the interlock
16 detection circuit, the register control circuit operable to

17 permit an instruction to access logical registers via the
18 corresponding physical data registers in the absence of
19 detection of a register access conflict,

20 stall an instruction a number of pipeline stages
21 indicated by the stall vector upon detection of a read after
22 write (RAW) register access conflict and thereafter permit the
23 instruction to access logical registers via the physical data
24 registers, and

25 write data to a shadow register upon detection of a write
26 after read (WAR) or write after write (WAW) register access
27 conflict and following expiration of a number of pipeline
28 stages indicated by the stall vector transfer data from the
29 shadow register to the physical data register corresponding to
30 the logical register of a data destination of the instruction.

1 22. (New) The digital data processor of claim 21, wherein:
 2 the set of shadow registers is less than a maximum number of
 3 instructions that can simultaneously cause register access
 4 conflicts of instructions between the pipeline stages; and
 5 the register control circuit is further operable to

6 write data to a shadow register upon detection of a write
 7 after read (WAR) or write after write (WAW) register access
 8 conflict if a shadow register is free and following expiration
 9 of a number of pipeline stages indicated by the stall vector
 10 transfer data from the shadow register to the physical data
 11 register corresponding to the logical register of a data
 12 destination of the instruction and free the shadow register,
 13 and

14 stall an instruction a number of pipeline stages
 15 indicated by the stall vector upon detection of a write after
 16 read (WAR) or write after write (WAW) register access conflict
 17 if no shadow register is free and thereafter permit the
 18 instruction to access logical registers via the physical data
 19 registers.

1 23. (New) A method of data processing in digital data
 2 processor including a processor pipeline with a plurality of
 3 pipeline stages and operable upon instructions specifying operands
 4 from a plurality of logical registers, the method comprising the
 5 steps of:

6 detecting write after read (WAR), write after write (WAW) and
 7 read after write (RAW) register access conflicts of instructions
 8 between the pipeline stages;

9 upon detecting a register access conflict generating a stall
 10 vector indicating a number of pipeline stages of stall for the
 11 detected conflict; and

12 in the absence of detection of a register access conflict,
13 permitting an instruction to access logical registers via a fixed
14 corresponding physical data registers;

15 upon detecting a read after write (RAW) register access
16 conflict, stalling an instruction a number of pipeline stages
17 indicated by the stall vector and thereafter permitting the
18 instruction to access logical registers via the physical data
19 registers; and

20 upon detection of a write after read (WAR) or write after
21 write (WAW) register access conflict, writing data to a shadow
22 register and following expiration of a number of pipeline stages
23 indicated by the stall vector transferring data from the shadow
24 register to the physical data register corresponding to the logical
25 register of a data destination of the instruction.

1 24. (New) The method of claim 23, wherein the set of shadow
2 registers is less than a maximum number of instructions that can
3 simultaneously cause register access conflicts of instructions
4 between the pipeline stages, the method further comprising the
5 steps of:

6 upon detection of a write after read (WAR) or write after
7 write (WAW) register access conflict if a shadow register is free
8 writing data to a shadow register and following
9 expiration of a number of pipeline stages indicated by the
10 stall vector transferring data from the shadow register to the
11 physical data register corresponding to the logical register
12 of a data destination of the instruction, and

13 freeing the shadow register; and

14 upon detection of a write after read (WAR) or write after
15 write (WAW) register access conflict if no shadow register is free

16 stalling an instruction a number of pipeline stages
17 indicated by the stall vector, and

18 thereafter permitting the instruction to access logical
19 registers via the physical data registers.